Use of Nested Certificates for Efficient, Dynamic and Trust Preserving Public Key Infrastructure

ALBERT LEVI
Sabanci University
and
M. UFUK CAGLAYAN
Bogazici University
and
CETIN K. KOC
Oregon State University

Certification is a common mechanism for authentic public key distribution. In order to obtain a public key, verifiers need to extract a certificate path from a network of certificates, which is called Public Key Infrastructure (PKI), and verify the certificates on this path recursively. This is the classical methodology. Nested certification is a novel methodology for efficient certificate path verification. Basic idea is to issue special certificates – called nested certificates – for other certificates. Nested certificates can be used together with classical certificates in Public Key Infrastructures (PKIs). Such a PKI, which is called Nested certificate based PKI (NPKI), is proposed in this paper as an alternative to classical PKI. The concept of "certificates for other certificates" results in nested certificate paths in which the first certificate is verified cryptographically while others are verified by just fast hash computations. Thus, we can employ efficiently verifiable nested certificate paths instead of classical certificate paths. NPKI is a dynamic system and involves several authorities in order to add a new user to the system. This uses the authorities' idle time to the benefit of the verifiers. We formulate the trade-off between the nested certification overhead and the time improvement on certificate path verification. This trade-off is numerically analyzed for a 4-level 20-ary balanced tree-shaped PKI and it has been shown that the extra cost of nested certification is in acceptable limits in order to generate quickly verifiable certificate paths for certain applications. Moreover, PKI-to-NPKI transition preserves the existing hierarchy and trust relationships in the PKI, so that it can be used for PKIs with fixed topology. Although there are many certificates in NPKI, certificate revocation is no more of a problem than with classical PKIs. NPKI even has an advantage on the number of certificate revocation controls: at most two certificate revocation controls are sufficient independent of the path length. Nested certificates can be easily adopted into X.509 standard certificate structure. Both verification efficiency and revocation advantage of NPKI and nested certificates make them suitable for hierarchical PKIs of wireless applications where wireless end users have limited processing power.

Categories and Subject Descriptors: K.6.5. [**Management of Computing and Information Systems**]: Security and Protection – Authentication; E.3 [**Data Encryption**]: Public key cryptosystems; C.2.0 [**Computer Communication Networks**]: General – Security and Protection

General Terms: Security, Performance

Additional Key Words and Phrases: Digital Certificates, public key infrastructure (PKI), nested certificates, key management

Earlier version of portions of the paper appeared in the proceedings of 2000 IEEE Symposium on Security and Privacy and IFIP TC11 16th International Conference on Information Security (IFIP/SEC'01). This work was supported in part by TUBITAK (grant EEEAG-237), Turkish State Planning Organization (grant DPT-96K120490).

Authors' addresses: Albert Levi, Faculty of Engineering and Natural Sciences, Sabanci University, Orhanli, Tuzla, Istanbul, 34956 Turkey; email: levi@sabanciuniv.edu; M. Ufuk Caglayan, Department of Computer Engineering, Bogazici University, Bebek, Istanbul 80815, Turkey; email: caglayan@boun.edu.tr; Cetin K. Koc, School of EECS, Oregon State University, Owen Hall 220, Corvallis, Oregon 97331; email: koc@ece.orst.edu.

1. INTRODUCTION

Public Key Infrastructure (*PKI*) is a general mechanism that provides authentic public key distribution to be used by large and distributed public key cryptography-based applications. A PKI is a certificate network used to find the correct public keys of users. A *certificate* is a digitally signed binding between a public key and one or more attributes of its owner. Those attributes can be the owner's identity such as name, e-mail address, URL (Uniform Address Locator), or authorizations that can be used to grant permissions or capabilities. Certificates are issued by trusted *Certification Authorities* (*CAs*). The verifier verifies the digital signature of the CA over the certificate and finds the correct public key for the certified user. However, in a PKI, there are several CAs and the verifier cannot know the public key of each CA. Therefore, the verifier spends most of its time to verify a *certificate path* with several certificates derived from PKI in order to find the public key of a user. The last certificate on the path is the certificate of the user whose public key is being sought. Each certificate on a path is verified to find the public key of the next CA and each public key is used to verify the next certificate. The verifier has to know the public key of the first CA in order to start this verification chain. Moreover, the verifier has to trust all CAs on the path regarding their honest behaviour and the legitimacy of their certifications. Such a trust is necessary, because otherwise the verifier cannot comment on the correctness of the information within the certificates issued by CAs.

Several PKIs are proposed in the literature. Most of them are based on the third edition of the ISO/ITU-T X.509 [ITU-T 1997] certificate standard. Privacy Enhanced Mail (PEM) [Kent 1993] is the first functional X.509 based system. It is intended to create confidential and authentic e-mail transfer between its users. Secure Electronic Transaction (SET) [MasterCard 1997] is yet another X.509 based system. The United States Postal Service (USPS) [1998] initiated the Information-Based Indicia Program (IBIP) to support new public key cryptographic methods for using postal services on the Internet. The proposed infrastructure for IBIP is a three level X.509 based hierarchical PKI. Public Key Infrastructure for X.509 certificates (PKIX) [Housley et al. 2002; Adams and Farrell 1999] is a general certificate infrastructure. Secure/Multipurpose Internet Mail Extensions (S/MIME) [Ramsdell 1999] is a secure Internet mail system. The certification infrastructure of S/MIME is based on PKIX infrastructure. Chokhani [1994] proposed an X.509 based national PKI. There are also non-X.509 based PKIs. Pretty Good Privacy (PGP) [Stallings 2003] e-mail security system seems to have the most widely used PKI of this category. The Simple Public Key Infrastructure (SPKI)

[Ellison et al. 1999], Simple Distributed Security Infrastructure (SDSI) [Rivest and Lampson 1996] and Domain Name System SECurity extensions (DNSSEC) [Eastlake 1999] are other examples of non-X.509 based PKIs.

Although the X.509 standard does not enforce any topology for a standard PKI, X.509 based PKIs are generally hierarchical and centralized. The general characteristics of an X.509 based PKI are (i) strict distinction between a CA and the end user (that is, the end users cannot issue certificates), (ii) a tree hierarchy with 3-7 levels and (iii) forming optional CA networks via cross certificates (not applicable for PEM). Moreover, the roles and responsibilities of the CAs in the specific levels are well defined in the PKI specifications and most of the time it is not possible to bypass a level in the hierarchy. PEM and SET are very strict on this issue.

Important for PKI and certification systems is the concept of "trust". The verifier must trust the CA in order to make sure about the legitimacy of the information given in a certificate. Although CAs are known as trusted entities, the verifiers must be able to choose their trusted CAs. In the X.509 based systems, every CA is a potentially trusted entity, but there are some mechanisms to avoid "blind trust" to the CAs. In the third edition of X.509 [ITU-T 1997], *policy identifiers* were added to the certificate structure as an optional extension. Those identifiers determine the certification practice of the CA. By checking a written policy, which is bound to the policy identifier in the certificate, the verifier can decide if that CA is trusted for issuance legitimate certificates. Policy management issues have been improved in the fourth edition of the X.509 standard [ITU-T 2000].

We call the certificate and PKI systems discussed above, X.509 based or not, "classical", as opposed to the "nested certification" system which is introduced below.

1.1. The focus of the paper

Classical certification systems use public key cryptography in order to digitally sign and verify the certificates. Public key cryptography operations are generally time inefficient. Moreover, all certificates on the path must be verified one by one in order to verify the certificate of a target user. The verifier only wants to find the correct public key of the target entity, but it has to verify the certificates of all intermediate CAs on the path and find their public keys in order to reach the target entity. We think that this is an unnecessary process degrading time efficiency.

One way of improving certificate path verification time is to let each CA verify the public keys of the end users via certificate paths and issue *direct classical certificates* for them. In this way, the verifiers who want to find the public key of an end user can quickly verify a direct classical certificate instead of following a certificate path. PGP [Stallings 2003] and ICE-TEL [Chadwick et al. 1997] use similar approaches. However, direct certification cannot be used for the distributed PKIs where the topology and the trust relationships must be preserved due to pre-established relationships among CAs at different levels.

In this paper, we propose a PKI, *Nested certificate based PKI* (*NPKI*), from which it is possible to extract efficiently verifiable certificate paths. Although NPKI is based on a new *nested certification* [Levi 1999] concept, both classical and nested certificates are used together. A nested certificate is simply defined as a certificate for another certificate. Certification authorities issue nested certificates for the certificates issued by their children in the PKI hierarchy. In this way, a classical PKI is transformed into an NPKI. Extra nested certificates are created during this process, but the initial topology of the source PKI is preserved in all stages of the transition process. That makes the NPKI model superior to direct classical certification. Our primary focus is on PKIs with a hierarchical backbone.

The certificate paths extracted from NPKI, namely *nested certificate paths*, can be verified by only one signature verification which is for the first certificate on the path. Other certificates are verified by hash computations. In this way, verification speed is increased tremendously as compared to classical PKIs and classical certificate paths. The cost of verification speed-up is the increase in the number of certificates in the system. In other words, there is a trade-off between verification speed and certificate issuance in NPKI.

NPKI should be seen as a system that provides an opportunity to manipulate the trade-off between the load on the end users (verifiers) and the load on the servers (CAs). NPKI enables to reduce the burden on the end users by increasing the load on the servers. That might be an advantage especially for wireless applications.

The intuition is that an increase in the number of certificates implies a bigger problem for certificate revocation in NPKI. This intuition is not correct. Certificate revocation rules are different for nested certificates and NPKI. In fact, NPKI is advantageous for certificate revocation. It is possible to have at most two certificate revocation controls per path independent of path length.

Adoption of nested certificates into the X.509 standard is analyzed and concluded that X.509v3 certificate structure that has been defined in the third edition of the standard

[ITU-T 1997] and remain unchanged in the fourth edition [ITU-T 2000] can be used for nested certificates too.

An overview of nested certification and nested certificate paths is found in Section 2. In Section 3, the transition from an existing PKI model is detailed. Certificate revocation characteristics, rules and advantages of nested certificates and NPKI are described in Section 4. Performance evaluation of the proposed method is given in Section 5. Moreover, the nested certification overhead is analyzed and the trade-off between this overhead and efficiency improvement is interpreted in this section. X.509 compatibility issues are discussed in Section 6. Nested certificates and other signed certificate validation mechanisms are compared in Section 7. The use of nested certificates in WAP security is the subject of Section 8. Section 9 gives the conclusions.

2. NESTED CERTIFICATION

In this section, the nested certificate structure, subject certificate verification method and the nested certificate paths will be explained. This section is an extract from Levi [1999].

2.1. Definitions and terminology

In simple terms, a nested certificate is defined as "*a certificate for another certificate*". A classical certificate gives assurance about the correctness of the binding between the identity and the public key of an entity. Therefore, it is verified to find the correct public key of the certified entity. A nested certificate, on the other hand, certifies another certificate by assuring the legitimacy of the signature over it. Therefore, nested certificates are used to verify the signatures over other certificates. For example, certificate 1 is a classical certificate in Figure 1, since it is issued by *A* to verify the public key of *B*. Certificates 2 and 3 are nested certificates in Figure 1, since they are issued to certify other certificates. Certificate 2 is issued by *C* to certify certificate 1. Similarly, certificate 3 is issued by *D* to certify certificate 2. *Nested Certification Authority (NCA)* is the authorized issuer of a nested certificate. For example, *A* is a CA, *C* and *D* are NCAs in Figure 1. The certificate, which is certified by a nested certificate, is called *subject certificate*. For example, certificate 1 is the subject certificate of nested certificate 2, in Figure 1. Similarly, certificate 2 is the subject certificate of certificate 3. Subject certificate is not a new certificate type. Any classical or nested certificate can be a subject certificate. For example, in Figure 1, the subject certificate 1 is a classical certificate and the subject certificate 2 is a nested certificate.

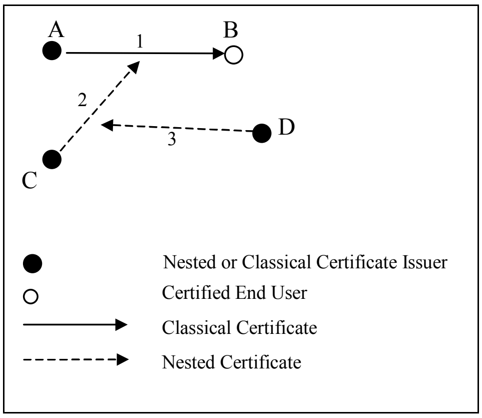

Fig. 1.The certification relationships

2.2. Structure and Issuance

An NCA issues a nested certificate by digitally signing the one-way hash of the nested certificate content. The content of a nested certificate is related to its requirements. The two requirements of a nested certificate are:

(i) to certify that the subject certificate content has been signed by the claimed CA or NCA and

(ii) to certify that the subject certificate content has not been maliciously modified.

In order to satisfy the first requirement, the nested certificate contains both the hash and the existing signature over the subject certificate content. The requirement is satisfied since these two values and the corresponding public-key are sufficient to verify the subject certificate signature. In order to satisfy the second requirement, the nested certificate contains the hash of its subject certificate content. This hash can be obtained by applying an irreversible one-way hash function [Rivest 1992; NIST 1995] to the subject certificate content.

Nested certificate issuance is basically two-step process.

1. In order to issue a legitimate nested certificate, the NCA must have verified the signature over the subject certificate content successfully beforehand. Both prior verification of the subject certificate and nested certificate issuance require computation of the hash of the subject certificate content. In order to have only one hash computation and to satisfy the first requirement described above, same hash algorithm is used for both operations.

2. Having verified the subject certificate, the NCA combines the hash value over the subject certificate content that is already computed, the existing signature over the subject certificate and some other fields, such as version numbers, algorithm identifiers, etc., into a single structure and digitally signs them in order to issue a nested certificate.

Formal representation of a nested certificate is given below. The notation used in this representation and the subsequent formal representations is given in Table I. Given a subject certificate SC = $\mathrm{Cnt}_{\mathrm{SC}}|\mathrm{Sig}_{\mathrm{SC}}$, a nested certificate for SC issued by *NCA* is denoted as:

$$\mathrm{NC}_{NCA}(\mathrm{SC}) = \mathrm{Cnt}_{\mathrm{NC}}|\mathrm{Sig}_{\mathrm{NC}} = \mathrm{Cnt}_{\mathrm{NC}}|NCA_s[\mathrm{H}[\mathrm{Cnt}_{\mathrm{NC}}]],$$

where $\mathrm{Cnt}_{\mathrm{NC}} = schash|scsig|Other$, $schash = \mathrm{H}[\mathrm{Cnt}_{\mathrm{SC}}]$, which is the subject certificate hash, $scsig = \mathrm{Sig}_{\mathrm{SC}}$, which is the subject certificate signature and *Other* is the other managerial fields such as algorithms used, serial numbers, etc.

Table I. The notation used in formal representations

Notation	**Meaning**
X_p	The public key of *X*
X_s	The secret (private) key of *X*
$X_s[\mathrm{I}]$	The signature of *X* over the information I. The signature is issued by X_s.
$X_p[\mathrm{I}]$	Inverse operation of $X_s[\mathrm{I}]$. Expected to return I', if $\mathrm{I}=X_s[\mathrm{I}']$.
H[I]	Hash (message digest) of information I.
$\mathrm{I}_1|\mathrm{I}_2$	Concatenation of the information I_1 and I_2.
$\mathrm{NC}_{NCA}(\mathrm{SC})$	The nested certificate, which is issued by *NCA*, for the subject certificate SC.
$\mathrm{Cnt}_{\mathrm{cert}}$	Content of certificate cert (does not include the digital signature).
$\mathrm{Sig}_{\mathrm{cert}}$	Signature over $\mathrm{Cnt}_{\mathrm{cert}}$.

It is extremely important to realize that the NCA does not guarantee the correctness of information of the subject certificate by issuing a nested certificate. The NCA guarantees the legitimacy of the signature over the subject certificate and conveys this information to the verifiers; no trust information is conveyed by a nested certificate. Therefore, in order to issue a nested certificate, the NCA need not trust anyone, even the subject certificate issuer, except itself. The NCA should only make sure about the legitimacy of the public key of the subject certificate issuer. This is a precondition for the step 1 of the nested certificate issuance process described above. There is a trust implication here such that the NCA should trust the CA of the subject certificate issuer. However, in the hierarchical PKI topology that will be detailed in Section 3.1, that CA is the NCA itself. Therefore in practice, the trust issue is only a self-trust in nested certificate issuance.

The fact that the nested certificate contains both the hash over the subject certificate content and the existing subject certificate signature based on this hash value may be considered as redundancy, since having a single hash over both the content and signature may work in a similar manner. However, in a single hash case, the nested certificate issuer has to compute different hash values in the prior verification of the subject certificate and in the nested certificate issuance. Moreover, having a single hash over the whole subject certificate is a violation of the requirement (i) given at the beginning of this section, because such a cumulative hash does not make possible to check whether or not the nested certificate issuer has issued a correct binding between the subject certificate content and the existing signature over it. The only advantage of having a single hash would be using less space in the nested certificate. In this paper, we preferred the nested certificates to contain separate subject certificate content hash and signature values. A detailed discussion on the single hash alternative can be found in Levi and Caglayan [1999a].

2.3. Cryptographic nested certificate verification method

In this method, the digital signature over the nested certificate content is verified by employing public key cryptosystem based signature verification operations. Different cryptosystems, such as RSA [Rivest, Shamir and Adleman 1978], DSA [NIST 1994] and ECDSA [Menezes 1993] can be used. Figure 2 gives the cryptographic nested certificate verification algorithm using RSA cryptosystem.

```
Given:  A nested certificate,
        $\mathrm{NC}_{NCA}\,(\mathrm{SC}) = \mathrm{Cnt}_{\mathrm{NC}}|NCA_s[\mathrm{H}[\mathrm{Cnt}_{\mathrm{NC}}]]$ issued by a
        trusted $NCA$ and the correct public key
        of $NCA$, $NCA_p$.

The verifier applies the following algorithm to verify NC.

Verified_Hash ← $NCA_p[NCA_s[\mathrm{H}[\mathrm{Cnt}_{\mathrm{NC}}]]]$
Calculated_Hash ← $\mathrm{H}[\mathrm{Cnt}_{\mathrm{NC}}]$
IF   Calculated_Hash = Verified_Hash   THEN
   NC becomes verified
ELSE
   NC has not been verified
```

Fig. 2. Cryptographic nested certificate verification algorithm using RSA

Verification of a nested certificate returns information about the correct hash value and signature over its subject certificate content. This information is used for the verification of the subject certificate using the *subject certificate verification* method.

2.4. Subject certificate verification method

The information returned by nested certificate verification is not sufficient to verify its subject certificate. By the verification of a nested certificate, only the correct hash value and correct signature over the subject certificate are found. In order to verify the subject certificate, the actual hash and the actual signature over the subject certificate must be compared with the ones stored in the nested certificate. Verification of a certificate as the subject certificate of a nested certificate is called *subject certificate verification*. Although the subject certificate verification method is an outcome of nested certification, it can be used to verify both nested certificates and classical certificates, since the subject certificates can be of both types.

Given: a subject certificate, SC = $Cnt_{SC}|Sig_{SC}$, issued by a trusted authority and a legitimate nested certificate for SC,
$NC_{NCA}(SC) = Cnt_{NC}|Sig_{NC} = Cnt_{NC}|NCA_s[H[Cnt_{NC}]]$,

where Cnt_{NC} contains the fields *schash*, which is equal to $H[Cnt_{SC}]$, and *scsig*, which is equal to Sig_{SC}.

The verifier applies the following algorithm to verify SC.

```
$Calculated\_Hash \leftarrow H[Cnt_{SC}]$
IF $Calculated\_Hash = Cnt_{NC}.schash$ AND
     $Sig_{SC} = Cnt_{NC}.scsig$ THEN
          SC becomes verified
ELSE
          SC has not been verified
```

Fig. 3. Subject certificate verification algorithm

The subject certificate verification algorithm is given in Figure 3. Having verified the nested certificate, *NC*, in order to verify its subject certificate, *SC*, the verifier follows two steps:

(a) The hash of the content of the actual *SC* is recalculated. This recalculated hash must be the same as the one stored within the *NC*.

(b) The actual signature over the content of the *SC* is compared with the subject certificate signature stored in the *NC*. These two signature values must be the same.

If the conditions given above are met and the verifier trusts the issuer of *SC* as an honest authority, then the verifier concludes that the *SC* is legitimate. The verifier must trust the issuer of *SC*, because the verification of *SC* does not mean that the information stored within *SC* is correct. The details of the trust issues will be given in Section 2.6.

The subject certificate verification method does not use public key cryptography operations. Therefore, it is more efficient than public key cryptography-based certificate verification. Moreover, this method has the same reliability as the cryptographic certificate verification method as shown in Levi and Caglayan [1999b].

A subject certificate can be another nested certificate. In this way, a nested certificate can be verified as the subject certificate of another nested certificate without using a cryptographic signature verification method.

2.5. Using nested certificates in PKIs and nested certificate paths

Nested certificates are not designed to replace all functions of the classical certificates, but to improve the performance and flexibility of them. Therefore, classical and nested certificates can be used together in PKIs and certificate paths. The PKI constructed in this way is called *Nested certificate based PKI* (*NPKI*). The certificate paths that are extracted from an NPKI are called *nested certificate paths*. A *nested certificate path* is a chain of nested certificates together with a classical certificate at the end. This classical certificate is for the *target entity* of the path. The eventual aim of using a nested certificate path is to certify the public key of the target entity. That is why the last certificate is a classical one. Each nested certificate of such a path, except the last nested certificate, is used to certify its subsequent nested certificate. The last nested certificate is to certify the classical certificate at the end.

A generic nested certificate path with k nested certificates (the certificates nc_k,nc_{k-1},nc_{k-2} … nc_3,nc_2,nc_1,cc_0) is shown in Figure 4. In order to verify such a path, the verifier must obtain all certificates on it and must know the public key of A_k, the first NCA of the path. The verifier must also trust all CAs/NCAs on the path in order to make sure about the correctness of the information within the certificates issued by them. The trust issues are discussed in more detail in Section 2.6. In a nested certificate path, each nested certificate is used to verify its subject certificate. At the end of a series of subject certificate verifications, the classical certificate, cc_0, of the target entity, T, is verified as the subject certificate of the last nested certificate, *nc_1*, of the path. Only the first nested certificate, *nc_k*, of a is verified cryptographically using the public key of its issuer, *A_k*. The other certificates of the path are verified as subject certificates. Therefore, the public keys of other nested and classical certificate issuers need not be found.

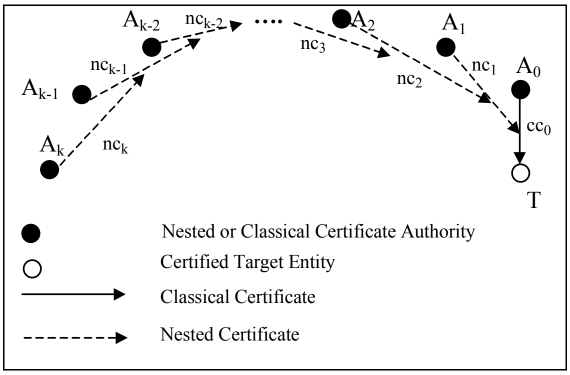

Fig. 4. A generic nested certificate path

Since the subject certificate verification method is more efficient than the cryptographic certificate verification method, the use of nested certificates on certificate paths improves the nested certificate path verification time as compared to the classical certificate paths of the same length. The speed-up factor for nested certificate path verification method ranges between 1.96 and 9.02 for nested certificate paths with 1 to 8 nested certificates as discussed in Section 5.2.

An NPKI, which is based on free certification model, was proposed in Levi and Caglayan [1998]. In this model, every CA is free to choose classical or nested certificates to issue. There is no enforcement. Therefore, this model suggests an organic growth from zero.

The contribution of this paper is to propose another NPKI model, which is called the *transition from an existing PKI* model. This model aims to convert an existing hierarchical PKI into NPKI. It allows nested certification as well as classical certification. There is systematic nested certification enforcement in this model. Every CA issues nested certificates to the certificates that are issued by its children in the PKI. In this way, it is possible to convert classical certificate paths into nested certificate paths without destroying existing hierarchical topology and trust relationships. The performance analysis shows that the use of nested certificates significantly improves the path verification time. However, there is a nested certificate issuance overhead for the authorities. The analyses given in Sections 5.3 and 5.4 show that this overhead is within acceptable limits for a medium size hierarchical PKI topology.

2.6. Trust issues

As discussed above, in order to verify a certificate, verification of the signature on the certificate is necessary. However, this verification is not sufficient without trusting the authority who signed the certificate. This trust is about the honesty of the CA or the NCA in the certification practice. In other words, it is the trust regarding the correctness of the information given in a certificate. Verification of a signature and the correctness of the content are two different concepts. Verification of a signature over a certificate is only the literal verification of the authority's signature. This verification does not mean that the content of the certificate is correct. That is why the verifier needs to trust the authority such that it really certifies correct information within certificates. A real world analogy can be the following. Suppose Alice sends a letter to Bob mentioning that Charlie owns a building in Manhattan. Bob may verify Alice's signature on that document since he knows how Alice's signature looks like, but that does not mean that Charlie really owns that building in Manhattan. Bob should also trust Alice in order to believe in the content of that document.

The above discussion can easily be generalized for a certificate path. In addition to the verification of the certificates, the verifier should have no doubts about the honesty of all CAs and NCAs on a classical or nested certificate path. The certificates on a path are verified (cryptographically, using subject certificate verification method, or both) in order to ensure that the signatures are legitimate. The verifier's trust in the authorities is necessary for making sure about the information correctness given in the certificate bodies.

The distinction between signature verification and trusting the authorities is not new for nested certificates. Classical certificates also have such a distinction. In classical certificate paths, the CAs, whose public keys are used in the signature verification process, must be trusted. Therefore, there is a close relationship between the signature verification and trust for classical certificate paths, although they are different concepts. On the other hand, this distinction becomes clearer in nested certificate paths and the subject certificate verification method. Although the subject certificate verification method does not involve the subject certificate issuer and its public-key, the verifier should trust the subject certificate issuer.

We do believe that trust establishment is more of a personal issue than being technical. The verifier should use his/her own judgment to assess the authority's credentials and to conclude about its trustworthiness after this personal evaluation. The certificate structure can only help the verifier by conveying authority's certification policy information. This is the technique employed by the X.509 certificates and briefly discussed in Section 1. A similar approach can be used for nested certificates as well.

3. NPKI CONSTRUCTION USING TRANSITION FROM AN EXISTING PKI APPROACH

This section discusses the construction of an NPKI using *transition from an existing PKI* approach. This approach deals with a systematic issuance of nested certificates throughout the infrastructure. Here, the goal is to have quickly verifiable nested certificate paths. In order to start nested certificate issuance, the CA systems should be updated so that they will be able to issue nested certificates as well as classical ones. Moreover, the verifier systems should also be aware of nested certificates in order to be able to verify nested certificate paths; that requires an update in the verifiers' side. Those updates constitute the initial step. After that, nested certificates are issued as explained in the subsequent sections.

3.1. Transition from PKI to NPKI

There is systematic nested certification enforcement in this approach. The CAs behave as NCAs. The nested certificates are node-to-arc arcs in NPKI. The method for the transition is called *nested certificate propagation*. This method is examined in two steps: 1) *Setup*, 2) *End user addition/deletion/update*. The *setup* step addresses the actual transition from the existing PKI into NPKI. The *end user addition/deletion/update* step addresses the cases where a new user is added, or an existing user is deleted, or a user's public key is changed. The classical certificates of PKI are neither deleted nor disabled in NPKI. However, they are not used if there is a nested certificate path to replace the classical certificates. The classical certificates must be kept since they may be needed before transition is finished or when a new end user is added.

Our design choice is to work on tree shaped hierarchical PKIs to convert them into NPKI. Hierarchical topologies are, actually, the real world cases for most of the applications. However, there might be some cross certificates that destroy the pure hierarchy. Thoughts on cross certificates will be explained below.

3.1.1. Nested certificate propagation – setup. The common practice in classical PKIs is to verify the certificate of an end user starting with a CA/NCA. The aim of the setup step is to obtain nested certificate paths only towards the end users. These paths can start with any CA/NCA from which there exists a classical certificate path in the original PKI.

The basic rule behind the ability of the nested certificate issuance in PKI to form an NPKI is as follows. Let A be an authority and A^c be the set of authorities that have been certified by A. A can validate the certificates, which had been issued by the authorities in A^c, since A already knows the public keys of them. Consequently, A can issue nested certificates for all certificates (nested or classical) that had been issued by the authorities in A^c. The above condition is necessary but not a sufficient condition.

In our method, the CAs create nested certificates for all cases that are in conformity with the rule given above, but there is an exception: CAs do not issue nested certificates for the classical certificates which belong to other CAs. However, the classical certificates, which belong to the end users, are certified. All nested certificates, if possible, are also certified. In this way, a nested certificate path is produced for each classical certificate path from a CA to an end user.

In the setup step, nested certificate issuance propagates from leaf nodes (end users) towards the root. Actually, the leaf nodes and their parents do not issue any nested certificate. First, nested certificates are produced by the CAs/NCAs, who are the grandparents of the leaf nodes. They issue nested certificates for the classical certificates of the leaf nodes. Then, these nested certificates are certified via other nested certificates issued by the parents of these CAs/NCAs. This propagating nested certificate issuance for nested certificates goes on until the root. At each iteration, each CA/NCA issues nested certificates for the nested certificates that are issued by its children.

An example setup phase over a tree-shaped PKI is given in Figure 5a and in Figure5b. First, certificates of the end users are certified via nested certificates in Figure5a. Then, these nested certificates are certified by the upper level root node in Figure5b. As can be seen in Figure5b, there is a nested certificate path towards each end user from all CAs/NCAs. However, as expected, there is no nested certificate path for the classical certificate paths between any two CAs/NCAs.

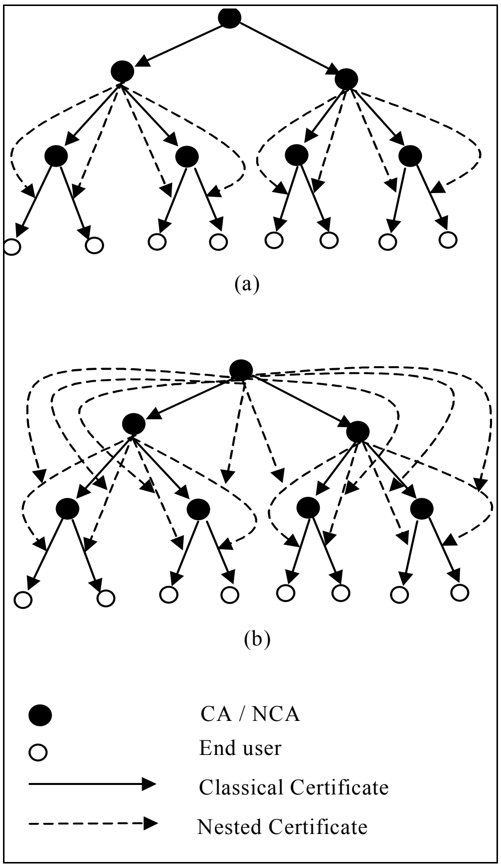

Fig. 5. An example transition to NPKI in two steps

Any possible cross certificate that destroys the hierarchy is not certified via a nested certificate in order to keep the number of nested certificates at a reasonable level. The paths that use those cross certificates still use the same cross certificates. However, nested certificate paths can be used before and after the cross certificates.

3.1.2. Nested certificate propagation – addition/deletion/update. A new user is added to a PKI via a classical certificate issued for it by a CA at upper-leaf level. The nested certificate propagation for this new user is the reduction of the setup algorithm for him/her. First, the classical certificate of the new user is certified by his/her grandparent CA/NCA with a nested certificate. Then, this nested certificate is certified via another nested certificate that is issued by the parent CA/NCA of the grandparent. This propagating nested certification goes on to the root such that, at each iteration, the nested certificate that is issued at the previous iteration is certified.

In practice, each iteration of nested certificate propagation requires a four-stage process. First, NCA should contact to the child CA/NCA to see if there are some subject certificates to be certified. Second, NCA should access the certificate storage in order to obtain those subject certificates. Third step is to verify those subject certificates and issue nested certificates as explained in Section 2.2. Fourth step is to store the newly issued nested certificates in the storage. Certificate storage and obtainment details are discussed in Section 3.2.8.

When a user is deleted, the classical certificate issued for this user is revoked. The revocation issues are discussed in Section 4.

To update the public key of a user, first a new entry with a new public key is added to the system. Then the previous entry is deleted. This order of operations should be preserved in order not to keep the user out of service during the update process.

3.2. Characteristics of NPKI and the nested certificate propagation method

Various characteristics of NPKI are discussed in the subsection. Comparison between nested certificate propagation and classical certificate propagation methods is also given here.

3.2.1. Efficient verification. The aim of the nested certificate propagation is to form nested certificate paths as alternative to the classical ones. The nested certificate path verification is faster than classical certificate path verification. These issues will be discussed in Section 5.

3.2.2. Dynamic certification. In classical PKIs, certification is a process between the issuer and the certified entity. However, certification affects several authorities in NPKI due to the nested certificate propagation method. This brings dynamism to the system such that the idle upper level authorities are involved in the lower level certification processes. As a result, the overall utilization increases. The nested certificate propagation method increases the load of the authorities but creates an opportunity to save time in the nested certificate path verification process.

In classical PKIs, upper-level authorities' private keys are activated infrequently mostly due to security reasons. The dynamism brought by involving upper-level authorities in certification processes in NPKI would remove this important characteristic of classical PKIs. The outcome of this fact of NPKI is discussed in the next section.

3.2.3. Availability requirements of the authorities. In classical PKI, once a CA, *C*, issues a classical certificate for an end user, *E*, some classical certificate paths towards *E* are automatically created. However, in order to create nested certificate paths corresponding to these classical certificate paths, nested certificate propagation is necessary. At first glance, it seems that authorities need to be available on-line during the propagation process, but this is not the case. Although the authorities must issue nested certificates for propagation, they need not accomplish this task just after the classical certificate issuance for *E*. The propagation process may be completed in time. The nested certificate propagation is carried out only to form efficiently verifiable nested certificate paths towards *E*. However, the absence of some nested certificates on these paths does not cause non-verifiability of *E*, since there are compensating classical certificates in the PKI. Only the efficiency improvement of the nested certificate path verification is not fully utilized for these cases; but this is temporary.

NCAs periodically check their neighbor CAs/NCAs in order to issue nested certificates for new certificates issued by the neighbors. The length of this period is related to the time for the completeness of the nested certificate propagation for a new user. If long time periods are chosen, the completion of the nested certificate propagation takes a long time. However, even if the nested certificate propagation is not complete, each nested certificate issuance improves the average certificate path verification time.

Another important point about nested certificate propagation is that the propagation method is sequential, not parallel. This implies that, each authority should wait for its children to finish nested certification in order to issue nested certificates. Therefore, the propagation delay is cumulative.

The availability requirements of the root-level CA deserve a close attention. In classical PKIs, root-CAs are off-line devices. Their private keys are strictly protected and activated infrequently under supervision. On the other hand, frequent root level private-key activations will be necessary when NPKI is adopted. NPKI is a system of trade-off between the authorities' time and the verifiers' time. In order to achieve fast verification, all authorities, especially the upper level authorities, must work harder than they work in a classical PKI. Since this requirement is an embedded characteristic of the proposed system, it is not possible to have a root CA in NPKI, which has exactly the same characteristics of a root CA in a classical PKI. However, as discussed above, CAs might still work in offline manner in NPKI. The frequency of private-key activations is another trade-off. Frequent activations may risk the security of the root CA's key, but enable speedy deployment of new users in NPKI. Infrequent activations may cause fewer security risks, but the deployment of newcomers will take more time. However, during this deployment period, classical certificate path or partial nested certificate path of a newly added user would make his/her certificate verifiable.

Frequent root-level CA key activations need not be automated. They may be performed with personal presence and audits, but the CA company should invest more on the human resources to perform this frequent off-line task.

3.2.4. Nested certificate propagation versus classical certificate propagation. An alternative approach to nested certificate propagation is the *classical certificate propagation* method, which produces *direct classical certificates*. Classical certificate propagation is theoretically possible in PGP [Stallings 2003] and ICE-TEL [Chadwick et al. 1997] systems. In this method, each CA verifies the paths towards the end users in the PKI starting with itself. Having verified each path, the CA issues a classical certificate for the verified end user. Using this method, it is possible to have only a single certificate between each CA - end user pair for which there was a classical certificate path in the PKI beforehand. The average certificate path (actually, single certificate) verification time for classical certificate propagation method is less than average certificate path (actually, nested certificate path) verification time for the nested certificate propagation method. This is an advantage of this method over nested certificate propagation. Moreover, the total number of nested certificates that must be issued in the nested certificate propagation method is equal to the total number of extra classical certificates that must be issued in the classical certificate propagation method. In addition, the total number of verifications that must be performed for propagation by each CA is the same in both methods. Why, then, must one insist on nested certificate propagation method, since it has no advantage over classical certificate propagation? Although the classical certificate propagation method is more advantageous, it is not possible to apply that method all the time. Some characteristics of the nested certificate propagation method make it preferable where classical certificate propagation is not possible:

1. Trust information-free certificate issuance.
2. Preservation of topology and trust relationships.
3. Suitability for distributed applications.

These characteristics will be discussed in Sections 3.2.5, 3.2.6, and 3.2.7, respectively. The cases where classical certificate propagation is not possible are also addressed these sections.

3.2.5. Trust information-free certificate issuance. In order to verify a public key through a classical certificate path, all intermediate CAs on the path must be trusted in the context described in Section 2.6. If even one of those CAs is not trusted, then the path cannot be validated. Therefore, in order to realize full classical certificate propagation, every CA of the PKI must trust everyone that can be reached starting with itself. Certainly, this is not a realistic assumption. On the other hand, partial classical certificate propagation is possible for the CAs, for which it is possible to verify some classical certificate paths that contain trusted CAs. The necessary condition for this case is that the direct classical certificate issuer must trust all intermediate CAs on the certificate path towards the target entity. However, if there is at least one untrusted CA on the path, then direct classical certificate issuer will not be able to verify the path and will not be able to issue a direct classical certificate for the target entity.

On the other hand, in order to issue a nested certificate for a subject certificate, the NCA does not need to trust anyone except itself – the NCA should trust the CA of the subject certificate issuer in order to make sure about the legitimacy of the public key of the subject certificate issuer, however that CA is the NCA itself in the hierarchical topology described in Section 3.1. Therefore, nested certificate propagation is applicable all the time, especially for the cases where classical certificate propagation is not applicable because of lack of trust. In order to issue a nested certificate, the NCA should only know the correct public key of the subject certificate issuer and should verify the signature over the subject certificate. The NCA does not need to validate other fields of the subject certificate and does not need to trust the subject certificate issuer or the entity certified within the subject classical certificate, because by definition, a nested certificate does not guarantee the information correctness of the subject certificate content. Besides the verification of a nested certificate and its subject certificate via that nested certificate, the verifier must also trust the subject certificate issuer and validate other fields of the subject certificate, in order to completely verify the subject certificate. In the nested certificate path verification, the verifier must trust all intermediate NCAs.

3.2.6. Preservation of topology and trust relationships. An implicit design decision of most PKIs is to let the verifiers decide on the trustworthiness of the authorities in the certificate path verification process. Moreover some PKIs, like PEM [Kent 1993] and SET [MasterCard 1997], assign fixed roles and responsibilities to the root and intermediate CAs. Similar situations may arise in international and interorganizational distributed PKIs as well. The topology and trust relationships in such PKIs should be preserved in such a way that the verifiers should follow the certificate paths enforced by the fixed topology.

Classical certificate propagation method spoils the existing trust relationships and the topology of the PKI, since the CAs enforce the verifiers to by-pass some CAs on the certificate paths by issuing direct classical certificates for the end users. This situation may cause a problem for PKIs with fixed topology, as described in the previous paragraph. In such PKIs, it is not possible to by-pass intermediate CA levels and consequently, classical certificate propagation is not possible.

In order to preserve the topology and the trust relationships, the authorities on the certificate paths, which the verifiers trust, must be the same authorities after the propagation. Nested certificate propagation method preserves trust relationships and the topology in this manner as described below.

In the subject certificate verification method, the verifiers should trust both nested and subject certificate issuers in order to verify a subject certificate via a nested certificate. Moreover, in the nested certificate propagation method, each authority issues nested certificates for the certificates that have been issued by its neighbors in order to create one nested certificate path for each classical certificate path of the PKI. A nested certificate path created in this way passes through the CAs that its corresponding classical certificate path also passes through. As a consequence, in the nested certificate path verification process, the verifier must trust exactly the same CAs of the corresponding classical certificate path. In this way, trust relationships within the PKI are preserved in the NPKI, which is constructed via nested certificate propagation. Moreover, since the trust relationships are preserved and there is no level by-pass in the nested certificate propagation method, it can be claimed that the topology of the original PKI is also preserved in NPKI.

For example, consider Figure 6. In this figure, both a classical certificate path, which is the certificate sequence cc_6, cc_5, cc_4, cc_3, cc_2, cc_1, cc_0, and its corresponding nested certificate path, which is the certificate sequence nc_6, nc_5, nc_4, nc_3, nc_2, nc_1, cc_0, are shown together. The verifier has to trust the authorities $A_{0,}$ A_1, A_2, A_3, A_4, A_5 and A_6 in order to verify both classical and nested certificate paths. Moreover, in both classical and nested certificate path verifications, the verifier must know the public key of A_6 a priori.

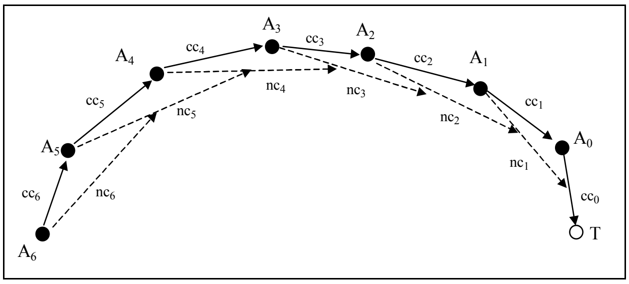

Fig. 6. An example classical certificate path and its corresponding nested certificate path

3.2.7. Suitability for Distributed Applications. The authorities should verify the paths starting from themselves in the classical certificate propagation method. Such an approach is centralized so that the direct classical certificate issuer should enumerate all classical certificates on the paths starting from itself. However, in the nested certificate propagation method, everyone is responsible for issuing nested certificates for the certificates that its certified nodes have issued. Moreover, in the nested certificate propagation method, the authorities need not enumerate paths to issue nested certificates. It is sufficient for an authority to check its certified authorities regularly for new certificates and issue nested certificates for these new certificates. This approach is more suitable for applications where certificates are stored and processed in a distributed manner, like DNSSEC [Eastlake 1999].

3.2.8. Certificate storage and obtainment. Certificate storage and obtainment in NPKI are not different from classical PKIs. One solution uses distributed directories, while the other has a centralized database. These solutions are explained below.

In NPKI, the certificates may be stored and obtained via distributed directories like X.500 directories [ITU-T 2001a] or the methods embedded in DNSSEC [Eastlake 1999]. The CAs/NCAs may serve as directory servers or they may publish the certificates that they issue to other directory servers. The verifier queries the directory to get the certificates on a nested certificate path to verify the public key of a specific end user.

The objects stored in directories should belong to entities with DNs (Distinguished Names). The directory users query the directory to search for objects belonging to a specific name. Classical certificates are such objects, but nested certificates are not, because they are issued for other certificates with no distinguished names. A possible solution to this problem is given below.

A common characteristic of nested certificates on a nested certificate path is that all of them can be used to verify only one classical certificate, which is at the end of the path.

This classical certificate is to verify the public key of an end user. Therefore, the identity of this end user is the common attribute of all certificates on a nested certificate path, and it can be used as the distinguished name for these certificates including the nested ones. However, since there can be several nested certificates on the path, the identity of the NCA of a nested certificate should also be considered for the sake of uniqueness. Moreover, the inclusion of the end user's identity in each nested certificate may facilitate the nested certificate path formation. The directory access mechanism can be improved so that the verifier is able to query the directory only once for all certificates containing the end user's identity. The returning certificates simply form a nested certificate path. Although directory mechanisms are commonly used especially for X.509 based PKIs, current systems should be upgraded in order to accommodate and support nested certificates as discussed above. In a largely deployed environment, such an upgrade is a major upheaval, but is inevitably necessary.

Another method of storing and obtaining certificates in NPKI may be to use a centralized database that contains all certificates in the system. In this method, as in the directory method, the nested certificates contain the identities of the end users of the corresponding nested certificate paths. Moreover, the database can be indexed by these end user identity fields. In that way, the retrieval process becomes faster and all certificates on a path can be obtained by a single query. However, as will be discussed in Section 5, the number of certificates in NPKI can be large. Therefore, having a single database may create storage capacity and corresponding efficiency problems.

4. CERTIFICATE REVOCATION AND RENEWAL

Classical certificates have limited lifespans, but CAs or certificate owners may need to revoke certificates before the expiration time. The reasons are given below.

- The private key corresponding to the public key in the certificate may be lost or compromised.
- The CA's signature key may be compromised.
- The certification contract may be terminated or the certificate holder's status and abilities described in certificate may change or may be cancelled (as by a person's leaving a job).

As discussed in Section 5, lots of extra nested certificates must be issued in NPKI. This would seem to cause an increased certificate revocation burden on top of the revocation of existing classical certificates in NPKI. Actually this is not correct. NPKI does not impose an extra certificate revocation load; NPKI requires fewer revocation controls for path verifications. This counter-intuitive fact is explained in this chapter. We start with a general introduction to existing revocation mechanisms for classical certificates and PKIs.

4.1. Certificate revocation in classical PKIs

Certificate revocation mechanisms must be incorporated into the PKI. The best-known revocation mechanism is the *Certificate Revocation Lists* (*CRLs*). A CRL keeps a signed list of the serial numbers of revoked certificates. Usually, the CA is the signer of the CRL for the certificates that it issued. A good discussion on CRLs can be found in Adams and Lloyd [1999].

Online Certificate Status Protocol (*OCSP*) is published as an RFC in Myers et al. [1999]. It is a simple request/response protocol that requires online servers, so-called OCSP responder, to distribute the certificate status on demand. Each CA must run its own OCSP responder, unless several CAs unite on this issue.

The literature contains other proposed methods of certificate revocation. Micali [1996] proposed the use of on-line/off-line signature scheme for a low-cost check for the "freshness" of a particular certificate. Naor and Nissim [2000] proposed authenticated data structures to represent CRLs. Kocher [1998] proposed Certificate Revocation Trees (CRTs). CRTs are used to compile the revocation information on a single hash tree. Gassko, Gemmell and MacKenzie [2000] proposed EFECTS (Easy Fast Efficient Certification System) that combines the best properties of certificates and CRTs. However, their system is best suited for a single CA issuing large numbers of certificates. Rivest [1998] proposed an agent based approach that employs on-line "suicide bureaus" to issue "certificates of health" for certificates. A recent certificate of health must be provided to the recipient along with the actual certificate. A brief taxonomy and overview of certificate revocation methods are given by Myers [1998].

CRLs, CRTs or the on-line revocation systems theoretically may become more centralized by having a single revocation authority to process all revocation data on behalf of CAs. Such an approach has the advantage of gathering all revocation information together, but it creates extra overhead in messaging among the CAs, certificate holders and the revocation authority. Moreover, several CAs must agree to delegate their revocation responsibility to the revocation authority. Therefore, central revocation authority is not suitable for distributed PKIs where CAs of different organizations interact.

Although there may be some exceptional cases where a single CA issues all certificates in a system, the PKI concept inherently employs a topology of several CAs. Therefore, the verifiers should verify a path of certificates in order to learn the public key of an end user. Consequently, they should check the revocation status of all certificates on the path. To do so the verifier needs to get the revocation information from all CAs on the certificate path. Thus, the difficulty of certificate revocation is multiplied by the amount of CAs (and certificates) on the path. NPKI has a certain advantage at this point, since only two revocation controls are enough whatever the certificate path length is.

4.2. Certificate revocation rules in NPKI

This section explains certificate revocation rules in NPKI. The implications of these rules will be discussed in the next section.

Rule 1: Classical certificates are revocable. The classical certificates for the leaf nodes of NPKI may be revoked, as in classical PKIs, if one of the situations discussed at the beginning of Section 4 arises. The guarantees and bindings given in these certificates are invalidated after revocation.

Rule 2: A revoked classical certificate makes its nested certificate path useless. The ultimate aim of a nested certificate path is to verify the classical certificate at the end. Moreover, a nested certificate can exist on only one nested certificate path. Therefore, when a classical certificate is revoked for some reason, all nested certificates on the nested certificate path towards it automatically become useless. Consequently, these nested certificates need not be revoked.

Rule 3: Do not start a nested certificate path with a revoked nested certificate, but revoked nested certificates can still be used on paths. If the key of a CA is compromised, then the nested certificates issued by it must be revoked, because these nested certificates must no longer be verified using the public key of the CA. However, this does not mean that these nested certificates contain bogus information. A nested certificate that has been issued for the revoked certificate before the revocation time can still be used for verification of the revoked certificate. For example, consider the example in Figure 7. Suppose the CA, *A*, has issued a nested certificate, *nc_1*, at time t_0. Later at time t_1>t_0, another CA, *B*, has issued a nested certificate, *nc*$_2$, for *nc*$_1$. At time $t_2 > t_1$, the public key of *A* is compromised and *nc*$_1$ is revoked. After t_2, it is not possible to verify *nc*$_1$ using the cryptographic method and the public key of *A*. However, it is still possible to verify *nc*$_1$ as the subject certificate of *nc*$_2$, which is still valid since *B* had issued *nc*$_2$ at time $t_1 < t_2$, i.e., before the revocation of *nc*$_1$. Moreover, *B* had verified *nc*$_1$ before issuing *nc*$_2$ and guaranteed the legitimacy of the signature over *nc*$_1$. The revocation of *nc*$_1$ at $t_2 > t_1$ does not invalidate the guarantee given by *nc*$_2$ at t_1.

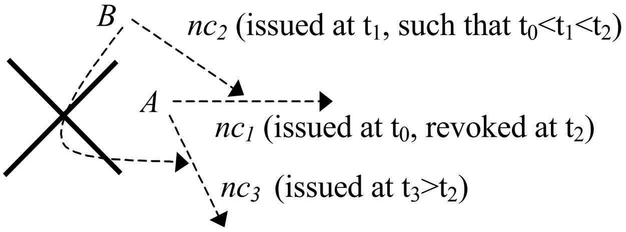
B
nc2 (issued at t1, such that t0<t1<t2)
A
nc1 (issued at t0, revoked at t2)
nc3 (issued at t3>t2)

Fig. 7. An example case for nested certificate revocation

On the other hand, the counterfeit of *A* can issue some bogus certificates (for example, *nc*$_3$ in Figure 7) at $t_3 > t_2$, i.e., after the compromise of its key. Since *B* and all other honest CAs will stop issuing nested certificates immediately after the revocation of *A*'s key, no nested certificates will be issued for *nc*$_3$ and for all other bogus certificates. Thus, the bogus certificates remain isolated and cannot exist on nested certificate paths, as long as they are not verified cryptographically as the first certificate of a path.

As a result, if the verifier finds a nested certificate path towards an end user, then it can safely verify the intermediate nested certificates[1] without any hassle for revocation control. Even if an intermediate certificate is revoked due to the revocation of its issuer's key, there already exists a nested certificate that certifies that the revoked certificate has been issued before the revocation of its issuer's key.

Moreover, no trusted time reference is necessary for NCAs and verifiers. An NCA abruptly stops nested certificate issuance when there is a problem with the key of its child. This implicitly guarantees the timely issuance of all previously issued nested certificates for the certificates issued by that child. Therefore, the verifiers do not need to check the time of issuance of a nested certificate; if a nested certificate exists within a path as an intermediate certificate, its timeliness is implicitly guaranteed. Revocation

[1] By "intermediate nested certificates", we mean all nested certificates on a nested certificate path excluding the first one. Revocation status of the first nested certificate on the path must be checked.

control for the first certificate of a path should be performed in anyway; so trusted time is not needed for the first certificate either.

Rule 4: No cascaded nested certificate revocations. A revoked nested certificate does not cause its subject certificate to be revoked. A nested certificate does not certify a public key or anything regarding a user. A nested certificate certifies only the relationship of the raw content of its subject certificate and the signature over it. The meaning of nested certificate revocation is that the CA of the nested certificate does not guarantee the correctness of the signature over the subject certificate anymore. However, the signature over the subject certificate can still be verified cryptographically using its issuer's public key. Therefore, nested certificate revocation is not a recursive process towards the end users.

4.3. Discussion on Certificate Revocation

The above rules imply that the verifier must check the revocation status of two certificates on a nested certificate path regardless of the path length. One of them is the first nested certificate, which is to be verified cryptographically. This certificate must be checked in order not to start the verification process with a bogus certificate (rule 3). The second certificate for which the revocation status must be checked is the last certificate of the nested certificate path, because it is a classical certificate and revoked classical certificates cannot be used (rule 1). Other nested certificates on the path need not be checked for revocation, because even if an intermediate nested certificate is revoked, this does not cause other certificates to be revoked (rule 4) and it can be used on the path since there exists another nested certificate on the path that had been issued to certify the revoked certificate before the revocation time (rule 3).

However, in a certificate path of a classical PKI, all certificates must be checked against revocation. Since all these certificates are from different CAs, different CRLs or OCSP responder contacts would be necessary for the revocation checks. Since there are only two certificate revocation controls in NPKI, the relative cost of certificate revocation decreases for the paths longer than two certificates as compared to classical PKI.

The revocation status of the first nested certificate of a nested certificate path must be checked since it might have been revoked due to a CA key compromise as discussed in rule 3. Revocation control can be waived if the verifier can make sure about the legitimacy and validity of the public keys that it uses to start the verification process. This would be possible by keeping CA key information in a local Personal Security Environment (PSE) and by periodically checking the validity of these keys. Similar approaches are proposed by PGP [Stallings 2003] and ICE-TEL [Chadwick et al. 1997] systems. However, the revocation status of the classical certificate at the end of a nested certificate path must always be checked.

One can argue that the CA compromise might go undetermined for a long time and during this period some bogus nested certificates might be disseminated. This is still not a major problem and does not require a mass revocation of innocent certificates. Once the breach is detected, it is sufficient to revoke the certificates issued by the compromised CA after the compromise, and the nested certificates issued on them recursively[2]. One may also argue that the counterfeit may change the timestamps in the certificates to make it seem they were issued earlier. This is not correct, because if the counterfeit does so, other CAs realize that something is going wrong and decline to issue nested certificates for the certificates issued by it. Thus, bogus certificates remain isolated.

The above discussion and rules 3 and 4 show that CA compromise in NPKI is not as severe as in classical PKI. Each CA controls its children by the nested certification process embedded in NPKI. There is no such control in a classical PKI. Once a classical certificate is issued, the issuer can no longer control the activities of the certificate holder.

Revoked certificates can be kept in Certificate Revocation Lists (CRLs) or handled by other methods cited in the beginning of Section 4. Each CA manages its own revoked certificates. There may also be nested certificates that are not revoked but are useless (rule 2). This situation inflates the databases/directories. A solution to this problem is to periodically run maintenance programs to locate and delete these useless nested certificates.

It is pretty clear that revocation control mechanisms in classical PKIs and NPKI are quite different. Having less number revocation controls in NPKI should not be considered as having a less reliable revocation control as compared to classical PKIs. In a classical certificate path, ultimate aim is to obtain the public key of the end user; intermediate CA certificates are verified and validated reluctantly. NPKI lets the verifier skip this reluctant process. Therefore, the difference between two revocation mechanisms should be seen as the removal of unnecessary intermediate steps for the same ultimate aim.

[2] If the verifier is concerned about the argument discussed in this paragraph, then he/she should check the revocation status of the first certificate of the path even if he/she makes sure about the validity of the public key of the corresponding CA, because this argument brings out a reason other than CA key compromise to consider a nested certificate revoked.

4.4. Nested Certificate Renewal

Classical certificates have limited lifespans to limit the misuse of a compromised private key corresponding to the public key in the certificate. Otherwise the revocation information for a revoked key must be kept indefinitely.

Since the nested certificates do not certify a public key directly, they do not need to bear expiration times. Nested certificates are bound to a classical certificate, which is at the end of the nested certificate path. Nested certificates on a nested certificate path automatically expire when the classical certificate that this path certifies expires. Expiration of that classical certificate makes all nested certificates on the path useless (a similar rule for revoked certificates has been given as rule 2 above). When the corresponding classical certificate has been renewed, all nested certificates on the nested certificate path must be reissued.

5. PERFORMANCE EVALUATION

Nested certificate path verification is more efficient than classical certificate path verification. However, a significant number of nested certificates must be issued to convert a PKI into NPKI and there is a trade-off between the number of nested certificates and efficiency improvement in certificate path verification.

Nested certification overhead, which is related to the number of nested certificates to be issued by the CAs/NCAs in NPKI, is analyzed for an hierarchical tree topology in this section. The trade-off between the nested certification overhead and the certificate path verification efficiency improvement is exemplified for an example case. The conclusion is that although nested certificate propagation improves the NPKI certificate path verification significantly, it increases the overhead costs, especially for the upper level authorities. Our analysis, which is given in the rest of this section, shows that for a 4-level hierarchical topology with 160,000 end-users, the overhead of nested certificate propagation is in acceptable limits. NPKI may be preferred for environments where the verifiers have limited capability for certificate path verification and the authorities are willing to work harder to the benefit of verifiers. Wireless environment is such an example since the wireless clients generally lack of processing power.

5.1. Nested certification overhead and trade-off analysis

Several nested certificates must be issued in the nested certificate propagation method. In this subsection, the nested certification overhead to convert the whole PKI into NPKI will be analyzed for a balanced tree shaped PKI/NPKI topology. Moreover, the trade-off between the nested certification overhead and the verification performance improvement will also be analyzed. Both performance figures are related to the number of certificates (both nested and classical) in the system. Therefore, the formulation will mostly be enumeration of certificates and paths.

5.1.1. Preliminaries. An *end user* is a user in PKI/NPKI, which does not issue certificates, but has certificates issued for it. The CAs/NCAs are the users of PKI/NPKI other than the end users. A *singular path* is a path with only one certificate. A *non-singular path* is a path with more than one certificate.

In the nested certificate propagation method, each CA/NCA, *A*, issues nested certificates for the certificates that are issued by the CAs/NCAs that *A* had certified. The only restriction to the above rule is that no nested certificates are issued for the classical certificates towards CAs/NCAs, since the aim of this method is to construct nested certificate paths towards only the end users.

The total number of nested certificates that must be issued by a CA/NCA is given by the next axiom.

Axiom 1: Suppose there is a PKI with the set of end users *E*. The total number of nested certificates that must be issued in order to form one nested certificate path for each classical certificate path between a CA/NCA, *A,* and the members of *E* is equal to the total number of distinct non-singular classical certificate paths between *A* and the members of *E*. These non-singular classical certificate paths may overlap, that is they may have common certificates.

The total number of nested certificates that must be issued in the global network is related to the total number of paths towards the end users from all of the CAs/NCAs in the PKI, as formalized by the following axiom.

Axiom 2: Suppose there is a PKI with the set of end users *E*. An NPKI is required to be constructed from this PKI such that there will be one nested certificate path for each classical certificate path towards the members of *E*. The total number of nested certificates that must be issued to attain this goal is equal to the total number of distinct non-singular classical certificate paths from all of the CAs/NCAs towards the members of *E*. These non-singular classical certificate paths may overlap, that is they may have common certificates.

5.1.2. Formulation for a Tree Shaped Topology. In this section, the *Nested certification overhead* and the *average nested certificate path length* will be formulated for a balanced tree shaped PKI/NPKI topology. Nested certification overhead is the factor of increase in the total number of certificates. This overhead value is always greater than or equal to 1. An overhead value 1 means that there is no overhead. An overhead value x means that nested certificate propagation increases the number of total certificates x times. The average nested certificate path length of the produced NPKI is also important in the analysis, since this value will give an idea of the efficiency gain for the PKI-to-NPKI transition.

Two nested certification overhead values will be formulated. One of them is the nested certification overhead to convert the whole PKI into NPKI, NCO_{PKI}, which is the ratio of the total number certificates (nested + classical) in the NPKI over the number of classical certificates in the PKI. The other nested certification overhead is for a single authority a. This overhead value, NCO_a, is the ratio of the total number of certificates (nested + classical) issued by a over the number of classical certificates issued by a. In order to formulate these overhead values, the number of nested certificates must be formulated. According to the axioms given in Section 5.1.1, the number of nested certificates is related to the number of paths. The number of paths is specific to the topology of the PKI and it is mostly not possible to formulate it for irregular graph shaped PKIs. In these graph shaped PKIs, it is also impossible to formulate average nested certificate path length. Therefore, one should work on regular PKI topologies. That is why a balanced tree topology will be used in the analysis. Such tree topologies are also common in real life PKIs. The topology that will be analyzed in this subsection is a k-level m-ary balanced tree.

A generic k-level m-ary balanced tree is shown in Figure 8. In a k-level m-ary balanced tree shaped PKI, each non-leaf node issues m classical certificates for their child nodes and there are k non-leaf node levels.

Let V_i represent the set of nodes in the level i. In a PKI, there are two sets of users. These are the set of end users and the set of CAs. These two sets are disjoint, that means they have no common members. In the PKI to be analyzed, the end users are the leaf nodes of the tree. Therefore, the end user set is denoted as V_k . The set of CAs contains other nodes of the tree. The members of the set of CAs will also act as NCAs in the NPKI.

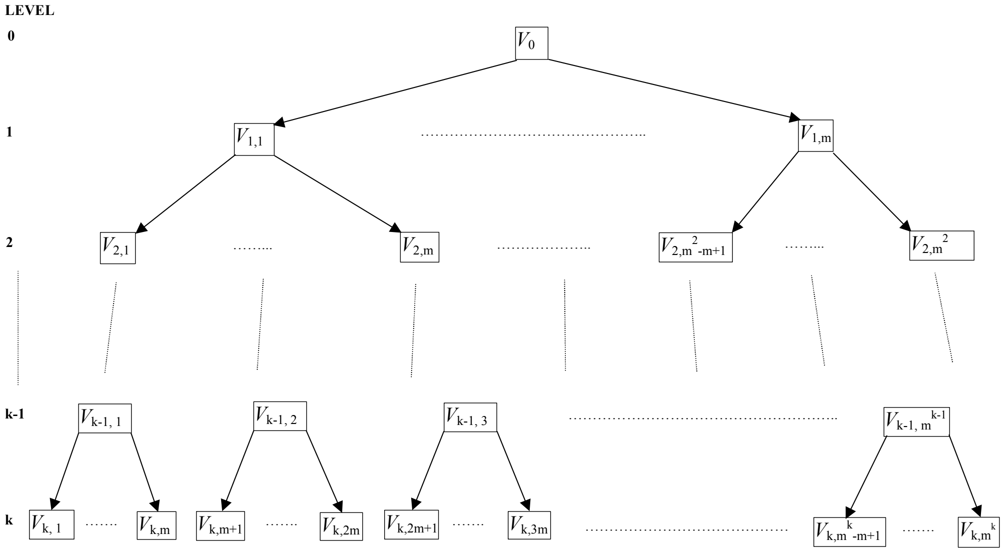

Fig. 8. A generic *k*-level *m*-ary balanced tree

Let $v_i \in V_i$. As axiom 1 implies, the total number of nested certificates from v_i towards the end users, which is denoted as n_{v_i}, is equal to total number of non-singular paths from v_i towards the end users. Since there are no non-singular paths from the nodes in V_{k-1} and V_k towards the end users, $n_{v_{k-1}}$ and n_{v_k} values are zero. n_{v_i} is formulated as follows.

$$n_{v_i} = \begin{cases} \prod_{j=i+1}^{k} m = m^{k-i} & i < k-1 \\ 0 & i \geq k-1 \end{cases} \qquad (1)$$

The number of nodes in the level *i*, which is denoted as $|V_i|$, is formulated as follows.

$$|V_i| = m^i \qquad (2)$$

According to Axiom 2, the total number of nested certificates that must be issued to convert the whole PKI into NPKI is equal to the total number of non-singular paths towards the end users. Moreover, each node at the same level shows the same characteristics in terms of nested certificate issuance. Under these considerations, the total number of nested certificates, which is denoted as n_{PKI}, is formulated as the following.

$$n_{PKI} = \sum_{i=0}^{k}\sum_{v \in V_i} n_v = \sum_{i=0}^{k} |V_i| \cdot n_{v_i} = \sum_{i=0}^{k-2} |V_i| \cdot m^{k-i} = \sum_{i=0}^{k-2} m^i \cdot m^{k-i} = \sum_{i=0}^{k-2} m^k = (k-1) \cdot m^k \qquad (3)$$

By definition, except for the leaf nodes (a.k.a. the end users), each node in the PKI issues *m* classical certificates. Under this consideration, the total number of classical certificates in the PKI, which is denoted as c_{PKI}, is formulated as the following.

$$c_{PKI} = \sum_{i=0}^{k-1} m \cdot |V_i| = \sum_{i=0}^{k-1} m \cdot m^i = \sum_{i=0}^{k-1} m^{i+1} \qquad (4)$$

Now the nested certification overhead values will be formulated. The nested certificate propagation overhead for a *k*-level *m*-ary balanced tree shaped PKI, NCO_{PKI}, is formulated as follows.

$$NCO_{PKI} = \frac{n_{PKI} + c_{PKI}}{c_{PKI}} = \frac{(k-1) \cdot m^k + \sum_{i=0}^{k-1} m^{i+1}}{\sum_{i=0}^{k-1} m^{i+1}} \qquad (5)$$

Let v_i be a node in the *i*th level of PKI and c_{v_i} be the number of classical certificates issued by v_i. c_{v_i} is equal to *m* for $\forall i \le k-1$ and equal to 0 for $i = k$. NCO_{v_i} is the nested certificate propagation overhead for v_i. NCO_{v_i} is formulated as the following.

$$NCO_{v_i} = \frac{n_{v_i} + c_{v_i}}{c_{v_i}} = \begin{cases} \dfrac{m^{k-i} + m}{m} = m^{k-i-1} + 1 & i < k-1 \\ 1 & i = k-1 \\ \text{n/a} & i = k \end{cases} \qquad (6)$$

As can be seen from this equation, for the end users, nested certification overhead is not applicable, since they do not issue certificates of any type. For the nodes in one upper level of the leaf nodes (level *k*-1), the nested certification overhead is 1. That is, there is no overhead, since they issue classical certificates, but do not issue nested certificates. For the upper levels, the nested certification overhead increases as the level number, *i*, decreases.

Nested certification overhead is the main disadvantage of the system. In order to compare this disadvantage with the advantage of efficient nested certificate path verification, the average nested certificate path length of NPKI must also be formulated. There is one path to each end user from each level. Therefore, there are *k* paths towards each end user. Moreover, the length of each path is only related to the level number. That is why the average nested certificate path length and the number of nested certificates values are solely dependant on *k*. The average nested certificate path length, which is denoted as pl, is formulated as the following.

$$pl = \frac{\sum_{i=0}^{k-1} k - i}{k} = \frac{\frac{k \cdot (k+1)}{2}}{k} = \frac{k+1}{2} \tag{7}$$

One certificate of a nested certificate path is a classical certificate. Therefore, the number of nested certificates on the average length nested certificate path is one less than pl. The number of nested certificates on the average length nested certificate, which is denoted as N, is then formulated as follows.

$$N = pl - 1 = \frac{k+1}{2} - 1 = \frac{k-1}{2} \tag{8}$$

5.2. Performance analysis of nested certificate path verification

The analytical and simulation based performance evaluation of the nested certificate path verification method is given in Levi [1999], Levi and Caglayan [1999b; 1999c]. The performance measure used in these analyses is the *speed-up* factor. The speed-up factor is the ratio of the classical certificate path verification time over the nested certificate path verification time. In Levi and Caglayan [1999b], eight sets of simulations are performed; each uses a different pair of public-key cryptosystem (RSA [Rivest, Shamir and Adleman 1978] or DSA [NIST 1994] with different key sizes) and hash algorithm (MD5 [Rivest 1992] or SHA-1 [NIST 1995]). We re-run those simulations for DSA-1024, RSA-1024 and RSA-2048. Paths with 1 to 8 nested certificates are considered in each set. Since there is one classical certificate at the end of each nested certificate path, there are 2 to 9 certificates total. The results for the simulations are shown in Figure 9. As can be seen from this figure, there is a remarkable improvement, especially for slower cryptosystems, like DSA-1024. For the cases considered, the speed-up factors are between 1.96 and 9.02.

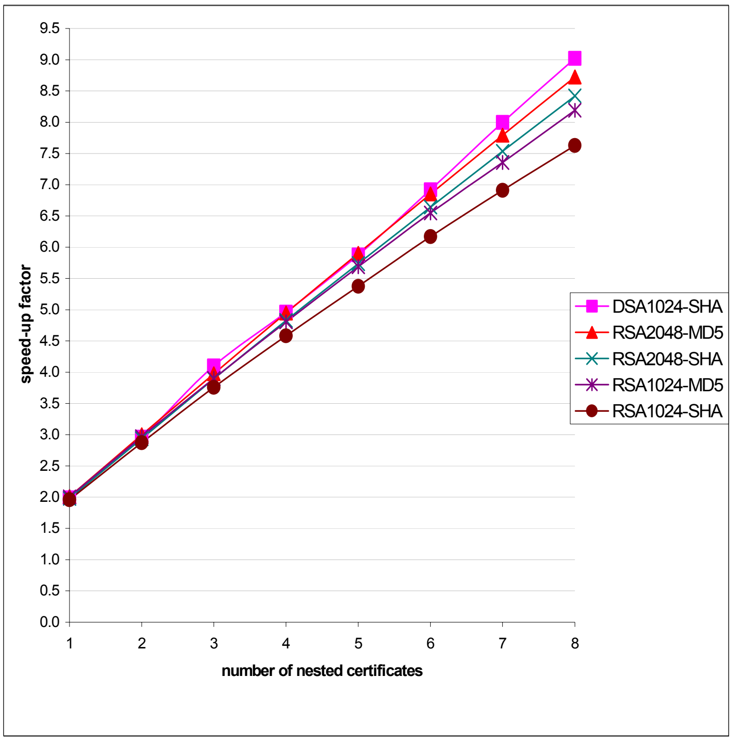

Fig. 9. Change of speed-up factor with respect to the number of nested certificates on the nested certificate paths

Table II gives nested certificate path verification speed-up factors for 3, 4 and 5 level trees. The speed-up factors are from Figure 9. *N*, the number of nested certificates on the average length path, is found using Equation 8. As can be seen from Table II, efficiency improvement gets better as the number of levels increases.

Table II. Speed-up factors for some hierarchical PKIs with different levels

Level (*k*)	**Number of nested certificates on average length path (*N*)**	**Speed-up factor range**
3	1	1.96 – 2.01
4	1.5	2.41 – 2.50
5	2	2.87 – 3.00

5.3. Numerical Overhead and Trade-off Analysis

In order to give an idea of the trade-off between the nested certificate path verification improvement and nested certification overhead, a numerical analysis is given in this subsection.

The change of NCO_{PKI} with respect to m is given in Figure 10 for 3, 4 and 5 level trees. Equation 5 is used for this figure. The primary factor effecting NCO_{PKI} is the number of levels (k). The behavior of NCO_{PKI} is asymptotic and approaches to k as m increases. Therefore, the nested certification overhead cannot exceed the number of levels.

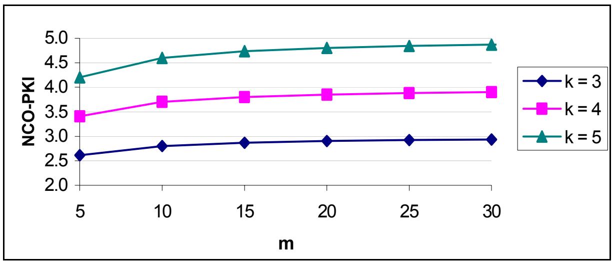

Fig. 10. Change of NCO_{PKI} with respect to m for different levels

NCO_{PKI} is the primary disadvantage of the system. On the other hand, it is also possible to have nested certificate paths that can be verified more efficiently. Therefore, there is a trade-off between efficiency improvement and NCO_{PKI}. For example, let us consider a 4-level, 20-ary balanced tree PKI (i.e., k = 4, m = 20). The NCO_{PKI} value for this PKI is found by applying Equation 5.

$$NCO_{PKI} = \frac{(k-1)\cdot m^k + \sum_{i=0}^{k-1} m^{i+1}}{\sum_{i=0}^{k-1} m^{i+1}} = \frac{(4-1)\cdot 20^4 + \sum_{i=0}^{4-1} 20^{i+1}}{\sum_{i=0}^{4-1} 20^{i+1}} = \frac{648420}{168420} = 3.85 \qquad (9)$$

For the same PKI topology, the nested certificate path verification speed-up factor is found between 2.41 and 2.50 on Table II. The trade-off is that the average path verification becomes 2.41 to 2.50 times faster by increasing the number of certificates 3.85 times. Although NCO_{PKI} is bigger than the efficiency improvement, this is not so bad, because the certificates are issued only once, but the verification can be performed several times.

Although NCO_{PKI} is low, the number of certificates is high. For the example PKI, the number of classical certificates is 168,420. In order to carry out nested certificate propagation on this PKI, a total of 480,000 nested certificates must be produced. However, these nested certificates can be produced in a few hours.

As discussed above, nested certification overhead is not evenly distributed among the NCAs in the hierarchy. However, the nested certification overhead is the same for the NCAs in the same level. The number of nested certificates that must be issued by NCAs,

n_{v_i} , is given by Equation 1. Level based Nested Certification Overhead, NCO_{v_i} , is given by Equation 6. Figures for the example 4-level, 20-ary balanced tree PKI are given in Table III. As expected, the upper level NCAs produce more nested certificates than the lower level NCAs. A discussion on the feasibility of this overload is given in Section 5.4.

Table III. Level by level number of nested certificates and number of NCAs for the 4-level 20-ary balanced tree PKI

Level (*i*)	**Number of nested certificates per NCA in the level** n_{v_i} (Eq. 1)	**Number of NCAs in the level** (m^i , Eq. 2)	**NCO in the level** NCO_{v_i} (Eq. 6)
0	160,000	1	8001
1	8,000	20	401
2	400	400	21
3	0	8,000	1 (no overhead)
4	0	160,000	Not applicable

5.4. Further discussion

An important criticism of the nested certificate propagation method may be the non-uniformity of the nested certificate issuance overhead. Indeed, a significant number of nested certificates must be issued by upper level CAs of the NPKI, as shown in Table III. In classical PKIs, the upper level CAs need not take any action, when a new end user is added to the PKI or the certificate for an end user is updated. However, in the nested certificate propagation method, when a new end user is added to the NPKI or the certificate for an end user is updated, one new nested certificate must be issued at each level. Although these characteristics seem to be disadvantageous, they can be justified:

1. Assuming that each CA/NCA stores the certificates that it has issued, the worst case (for the top level CA/NCA) storage requirement of the nested certificates for the 4-level 20-ary balanced tree case is in the order of 10 Mbytes, which is quite acceptable.
2. The nested certificates are issued once, but they are used to verify nested certificate paths several times. Therefore, the overhead is once, but the gain is several times.
3. The classical certificates of the PKI still exist in the NPKI. Therefore, there are always the classical backups of the nested certificate paths in NPKI. Thus, nested

certificate propagation can be considered as an “off-line” process. The authorities may issue nested certificates at idle times. During the initial set up or when a new end user certificate is issued, the verifiers may use the classical certificates until the nested certificate propagation is completed.

4. Classical CA servers are mostly idle and their utilization is small, since their classical certification loads are not so significant and they must be dedicated servers for security reasons. The utilization of these servers increases by the nested certificate propagation method.
5. Although the nested certificate issuance overhead seems to be significant, the example NPKI given above can be set up in a few hours without interfering with normal PKI operations. Such a setup time is acceptable. This setup time tends to increase for bigger PKIs and the nested certificate propagation method may become useless. For example, the set up time for a 5-level 30-ary balanced tree shaped PKI is 2-3 weeks. For such large PKIs, the bottleneck is mostly due to the top level CA, since it must issue large numbers of nested certificates itself. For these cases, nested certificate propagation can be considered for subhierarchies. That means, the top level CA does not issue nested certificates, but the CAs of its subhierarchies do. This approach can be recursive such that if the subhierarchies are also big, the nested certificate propagation is applied for their subhierarchies.

6. X.509 COMPATIBILITY

In X.509 based PKIs, certificate structures and certificate path processing rules are well defined by the X.509 standard [ITU-T 1997; 2000]. Use of nested certificates definitely changes certificate path processing rules in NPKI, so in order to adopt nested certificate paths in the X.509 standard, the standard needs some editing. However, certificate structure need not change. The version 3 certificate structure (X.509v3) defined in the third edition of the standard [ITU-T 1997] and unchanged in the fourth edition [ITU-T 2000] can be used for nested certificates too. However, since the interpretation of some certificate fields and, especially, certificate and certificate path processing are different in nested certificates, nested certificate compliant applications will need some significant modifications too. The processing details are explained in the previous sections. In this section, we will describe how the X.509v3 certificate structure can be used to accommodate nested certificates.

6.1. Different Attributes for the Certified Entity

The most important difference between a classical certificate and a nested certificate is the difference in content. A classical certificate certifies the binding between the name and the public key of an entity. Therefore, it contains a name and a public key as the attributes of the certified entity. On the other hand, a nested certificate certifies the binding between a certificate and its claimed signature. Thus, it contains a certificate identifier, instead of a name, and a hash and signature combination, instead of a public key, as the attributes of the certified entity. Below we analyze the representation of these attributes of the nested certificates in X.509v3 certificate structure.

- **Representation of the hash and signature:** In classical X.509v3 certificates, the public key of the certified entity is stored as a bitstring in the **subjectPublicKeyInfo** field of the certificate structure. The counterpart of this public key is a hash and signature combination in nested certificates. This combination is also bitstring, and the same **subjectPublicKeyInfo** field could be used for it. Therefore, there is no structural problem for the representation of the hash and signature combination. Once the type of the certificate (classical or nested) is identified as described in Section 6.2, the **subjectPublicKeyInfo** field could be interpreted either as the public key of the certified entity (for classical certificates) or as the hash and signature combination of the subject certificate (for nested certificates). It is obvious that the name **subjectPublicKeyInfo** does not carry the proper meaning for nested certificates. Therefore, in the next version of X.509 certificate structure this field could be renamed as **certifiedObject** in order to carry a general meaning for both classical and nested certificates, but this modification is not structurally necessary.
- **Representation of the subject certificate identifier:** There are several ways to represent the name of a classical certificate holder in the X.509v3 certificate structure. The **subject** field can be used for an X.501 [ITU-T 2001b] directory name, or the Subject Alternative Name (**subjectAltName**) extension can be used for more common identities like an e-mail address or URL. The counterpart of this certificate holder's name in nested certificates is the subject certificate identifier, which consists of two parts to uniquely identify the subject certificate: (1) the serial number of the certificate and (2) its issuer's name. Serial number is of type integer. We cannot use the **subject** field for serial number since their types mismatch. Moreover, neither of the **subjectAltName** choices is of type integer, but we can still use the **otherName** choice of **subjectAltName** extension for the serial number of the subject certificate since a generic type identifier can be assigned to it. The subject certificate issuer

name can be stored either in the **subject** field or as an extension in **subjectAltName**. In X.509 certificate structure, **subjectAltName** can be a list of names, so there is no problem for using that extension for both serial number and issuer name. However, there must be a convention about the order of the names for proper processing; for example, the serial number may always be the first entry in the list.

6.2. The Problem of Differentiation in the Certificate Types

In order to use the methods described in Section 6.1, the certificate type must be identified a priori. In order to identify whether a certificate is a classical or a nested one, Extended Key Usage (**extKeyUsage**) extension field of the X.509v3 certificate structure can be used. That field is normally used to assign an extra function to the certified public key of a classical certificate via object identifiers, but it can also be used to identify a certificate as a nested one. A publicly known object identifier can be assigned as the value of **extKeyUsage** field for the nested certificates, and the certificate processor checks this value to find out whether a certificate is nested or not.

6.3. CA – NCA Differentiation and Trust Considerations

A single authority can behave both as a CA and as an NCA in NPKI. Moreover, the verifier may trust an authority differently as a CA and as an NCA. Since the trust information is kept as policy identifiers in X.509v3 certificates, there must be different policy identifiers for the CAs and NCAs. This does not require a modification in the certificate structure, because the policy identifiers are assigned at the issuance time.

Needless to say, CA related fields of an X.509v3 certificate must be perceived as NCA related fields for nested certificates. Once the type of the certificate is identified as described in Section 6.2, this perception problem becomes an issue of implementation. Therefore, it is not necessary to modify CA related fields in the X.509v3 certificate structure.

7. NESTED CERTIFICATES VS. OTHER SIGNED CERTIFICATE VALIDATION MECHANISMS

Although the idea behind NPKI and nested certification is to improve the certificate path verification time, the proposed structure allows validation of certificates where this characteristic is utilized in revocation control. There are similar validation mechanisms in the literature. In this section, such mechanisms are explained and compared to NPKI and nested certificates.

OCSP (Online Certificate Status Protocol) [Myers et al. 1999] is defined as an online mechanism for certificate revocation control. It has a simple client-server architecture. The OCSP server (responder) processes the client requests of checking certificate revocation status and sends back a signed response to the client. This response contains certificate identifiers and their status.

A similar approach is employed by XKMS (XML Key Management Specification) [Ford et al. 2001]. XKMS uses web-based trust services as an interface to PKI and the clients talk to this service using simple XML transactions. In this way, the clients make use of PKI functionalities without actually dealing with PKI. As a part of this framework, tier-2 validation service of XKMS allows the clients to query the status of a binding between a public-key and its associated attributes. The trust service first uses its interface to PKI to check the status of the corresponding certificate and responds to the client with a signed XML response[3].

The basic difference between nested certificates and OCSP/XKMS validation mechanisms is at the purpose of the applications. OCSP and XKMS use validation mechanisms to check the validity status of certificates. This is not the primary aim in nested certificates; there are only some advantages in revocation control, which is basically a side-product. The primary aim of using nested certificates in NPKI is to be able to extract efficiently verifiable certificate paths. Moreover, OCSP and validation service in XKMS are not directly embedded in the PKI, so they do not take the challenge of changing the existing PKI and certificate standards. On the other hand nested certificates take that challenge and are embedded in the PKI. However, nested certificates are not designed to replace classical certificates, but to live together in symbiosis. In that sense, nested certificates and the validation mechanisms in OCSP and XKMS are all supporting measures to improve the effectiveness of PKIs, but from different perspectives.

Another certificate validation mechanism exists in SDSI (Simple Distributed Security Interface) [Rivest and Lampson 1996] and SPKI (Simple Public Key Infrastructure) [Ellison et al. 1999]. Those mechanisms do not have certificate revocation concept at all. Each certificate is assigned an appropriate validity period. The certificate times out after this period and needs *revalidation*[4]. Revalidations are performed either by the certificate issuers or by specific revalidation authorities. In order to revalidate a certificate, the

[3] XKMS specification does not enforce using digital signatures in response. If the link between the client and trust service is authenticated by some means (like SSL), then a digital signature is not necessary.

certificate issuer re-signs the certificate content with a new timestamp. However, revalidation authorities sign the whole certificate content and the original signature on it, in order to revalidate a certificate. The latter mechanism is very similar to nested certificate issuance, but the purpose is different as in the cases of OCSP and XKMS. In SDSI and SPKI, the aim is to revalidate the certificates to eliminate the revoked ones. However, this is not the primary aim in NPKI as discussed above.

8. COMMENTS ON USE OF NESTED CERTIFICATES FOR WIRELESS ACCESS PROTOCOL

Certificate based solutions became important for wireless systems security after the WAP (Wireless Access Protocol) [WAP 2001a] proposal. WAP is the wireless version of HTTP. Its aim is to provide a convenient Internet access for wireless devices. Two security protocols of WAP, namely WTLS (Wireless Transport Layer Security) [WAP 2001b] and WPKI (Wireless PKI) [WAP 2001c], extensively use certificates. WTLS and WPKI are *optimized* wireless versions of their wired counterparts TLS and PKI. They are optimized for both performance (for time considerations) and size (for bandwidth considerations). Certificate sizes are reduced in WTLS and WPKI. Fast cryptographic algorithms, like elliptic curve cryptography (ECC) [Menezes 1993], and smaller key sizes are preferred for faster processing. WTLS and WPKI standards offer 160-bit ECC curves and RSA-1024. Those key sizes provide moderate security. These performance considerations are for wireless end users only; wired servers can still perform bandwidth and computation intensive jobs.

Industrial applications on WTLS and WPKI certificates have a tendency towards a *flat* PKI in which a single CA issues certificates for all users, and there are no intermediate CAs and CA-to-CA certificates. In this way, each certificate path would contain a single certificate. This implies the following three advantages.

- Less bandwidth: wireless verifiers need to obtain a single certificate.
- Fast verification: wireless verifiers verify only one certificate.
- One revocation control, since there is only one certificate on the path.

Classical CRL and/or OCSP can still be used for the certificates of wireless end-users, since those certificates are validated by wired servers in WAP environment. On the other hand, revocation control for *server* certificates is problematic since wireless end-users are supposed to check those certificates. Even with a flat PKI, the wireless users should

[4] "Revalidation" is the term used by SPKI. SDSI uses "reconfirmation" instead of "revalidation"

obtain a CRL or contact to an OCSP server if classical revocation control mechanisms are employed. Verisign Inc's approach for server certificate revocation control is to employ *short-lived certificates*. A server certificate has a short lifetime, say one day, so they need not be revoked. The servers should periodically download and install fresh certificates. The burden is on the CA and on the server, but not on the wireless clients.

Although they are advantageous, Flat PKIs are not practical for distributed environments where there are several CAs. Use of nested certificates in WAP would help to form a multi-CA hierarchical PKI with fast verification time and advantageous certificate revocation control mechanisms.

As discussed in Section 5, nested certificate path verification is a fast process. It is possible to have some extra layers in the PKI hierarchy without a significant processing cost on the verifiers. Basically the cost of one extra layer is one extra hash computation for the wireless verifiers. Here the burden is on the wired CA servers. They spend extra time to issue nested certificates. The wireless verifiers benefit from those nested certificates by verifying the paths faster.

It is also possible to have single revocation control in nested certificate paths as discussed in Section 4. The same thing applies if a nested certificate based PKI structure is used for server certificates in WAP environment. The wireless end user obtains a nested certificate path towards a server certificate. Then it checks the revocation status of only the server's certificate which is at the end of the path, provided that the verifier makes sure about the legitimacy of the key of the first CA on the nested certificate path.

Even with the nested certificate paths, the revocation control of the single certificate (which is at the end of the path) is still a problem. The concept of short-lived certificates can also be used in this nested certificate based structure. However, the corresponding nested certificate path must be reconstructed after each certificate renewal. That would put an extra burden on CAs, but this approach is still feasible for medium size PKIs if the certificate renewals are evenly distributed in time. That would help the CA servers to use their time more effectively.

The nested certificate based solution for a hierarchical WAP PKI does not address the bandwidth problem. The wireless end users should still download all certificates on the nested certificate paths.

9. CONCLUSIONS

Nested certification and the corresponding subject certificate verification methods were proposed to improve certificate path verification times. A nested certificate is basically a certificate for another certificate. By using nested certificates in certificate paths, it is possible to have efficiently verifiable nested certificate paths. In this paper, the design of the *Nested certificate based PKI (NPKI)*, which incorporates both classical and nested certificates, has been presented. The NPKI construction model discussed in this paper is called *transition from existing PKI* and the method to realize the transition is called the *nested certificate propagation* method. This model enforces the CAs to issue nested certificates to the certificates, which are issued by their child CAs in the PKI. The outcome of this model is a nested certificate path for each classical certificate path towards the end users in the PKI. On a nested certificate path, all certificates are nested certificates except the last one. Therefore, the verification time of a nested certificate path is considerably less than a classical certificate path.

Another advantage of the nested certificate propagation method is that the trust structure and the topology of the PKI are not spoiled. Therefore, NPKI preserves trust. Moreover, NPKI is dynamic since several authorities are involved in the certification of an end user in a distributed manner. In this way, the utilization of the authorities increases to the benefit of the verifiers.

To attain verification time improvement, numerous nested certificates must be issued. That means there is a trade-off between the verification improvement and the nested certificate issuance overhead. This trade-off was also analyzed in this paper by using a generic balanced tree PKI model. It has been observed that, for a 4-level, 20-ary balanced tree shaped PKI, the average path verification speed-up factor is between 2.41 and 2.50, depending on the cryptosystems and the hash algorithms used. However, the number of total certificates increases by 3.85 times. Unfortunately, this certification overhead is not uniformly distributed among the authorities. Upper level authorities perform more nested certifications than the lower level ones. Despite this certification overhead, transition to NPKI can be performed in a few hours for the example hierarchy. Such an overhead may be tolerable in order to improve the path verification time in applications for which verification efficiency is more important than the burden on the authorities.

The increase in the number of certificates may make people think that certificate revocation is a bigger problem in NPKI. This is not correct. Since nested certificates are not for users, but for other certificates, revocation rules are different for them. Those rules show that at most two revocation controls are sufficient for nested certificate paths, independent of the path length. This number can be reduced to one for special cases, and it is independent of the path length. On the other hand, all certificates on a classical certificate path must be checked against revocation. Therefore, NPKI has a certificate revocation advantage over classical PKIs.

Compatibility between nested certificates and the X.509 standard [ITU-T 1997; 2000], is analyzed. Nested certificate structure can be easily adopted into X.509v3 standard certificate structure without any structural modifications. However, the standard certificate issuance and certificate path processing rules must be updated in accordance with nested certificate issuance and nested certificate path processing rules.

The use of certificates for wireless communication presents some challenges; certificates should not put extra processing and revocation control burden on wireless end users. Certificate use in the well-known Wireless Application Protocol (WAP) has a tendency towards *flat* PKI, where there is only one CA and consequently one certificate on each path. This tendency is not good for distributed application and for competition. Nested certificates would help to have *hierarchical* PKI for WAP. Nested certificate paths are fast enough to be verified by the wireless end user. Certificate revocation advantage of nested certificate paths also helps the end users.

ACKNOWLEDGMENTS

The authors are grateful to Audun Jøsang, John Burt and anonymous referees for their careful review and comments on this paper.

REFERENCES

ADAMS, C., AND FARRELL, S. 1999. *Internet X.509 Public Key Infrastructure Certificate Management Protocols*, RFC 2510.

ADAMS, C., AND S. LLOYD, S. 1999. *Understanding Public Key Infrastructures*. New Riders Publishing.

CHADWICK, D. W., YOUNG, A. J., AND CICOVIC, N. K. 1997. Merging and Extending the PGP and PEM Trust Models – The ICE-TEL Trust Model. *IEEE Network 11*, 3 (May/June), 16-24.

CHOKHANI, S. 1994. Towards a National Public Key Infrastructure. *IEEE Communications Magazine 32*, 9 (Sept.), 70-74.

EASTLAKE, D. 1999. *Domain Name System Security Extensions*, RFC 2535.

ELLISON, C., FRANTZ, B., LAMPSON, B., RIVEST, R., THOMAS, B., AND YLONEN, T. 1999. *SPKI Certificate Theory*, RFC 2693.

FORD, W., HALLAM-BAKER, P., FOX, B., DILLAWAY, B., LAMACCHIA, B., EPSTEIN, J., AND LAPP, J. 2001. *XML Key Management Specification (XKMS)*, latest version is available at http://www.w3.org/TR/xkms/

GASSKO, I., GEMMELL, P. S., AND MACKENZIE, P. 2000. Efficient and Fresh Certification. In *Proceedings of Public Key Cryptography (PKC) 2000* (Melbourne, Australia, January 2000), LNCS 1751, Springer-Verlag, 342 -353.

HOUSLEY, R., POLK, W., FORD, W., AND SOLO, D. 2002. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, RFC 3280.

ITU-T. 1997. Recommendation X.509, ISO/IEC 9594-8, *Information Technology – Open Systems Interconnection - The Directory: Authentication Framework*, Third Edition.

ITU-T. 2000. Recommendation X.509, ISO/IEC 9594-8, *Information Technology - Open Systems Interconnection - The Directory: Public-key and Attribute Certificate Frameworks*, Fourth Edition.

ITU-T. 2001a. Recommendation X.500, ISO/IEC 9594-1, *Information Technology - Open Systems Interconnection - The Directory: Overview of Concepts, Models and Services*.

ITU-T. 2001b. Recommendation X.501, ISO/IEC 9594-2, *Information technology – Open System Interconnection – The Directory: Models*.

KENT, S. T. 1993. Internet Privacy Enhanced Mail. *Communications of the ACM 36*, 8 (Aug.), 48 - 60.

KOCHER, P. 1998. On Certificate Revocation and Validation. In *Proceedings of Financial Cryptography 98* (Anguilla, BWI, Feb. 1998), LNCS 1465, Springer-Verlag, 172 -177.

LEVI, A. 1999. *Design and Performance Evaluation of the Nested Certification Scheme and its Applications in Public Key Infrastructures*, Ph.D. Thesis, Bogazici University, Dept. of Computer Engineering.

LEVI, A., AND CAGLAYAN, M. U. 1998. NPKI: Nested Certificate Based Public Key Infrastructure. In *Advances in Computer and Information Sciences '98 - Proceedings of the Thirteenth International Symposium on Computer and Information Sciences - ISCIS XIII* (Antalya, Turkey, Oct. 1998), Concurrent Systems Engineering Series 53, IOS Press, 397 - 404.

LEVI, A., AND CAGLAYAN, M. U. 1999a. Integrity Control in Nested Certificates. In *Proceedings of BAS'99, The Fourth Symposium on Computer Networks* (Istanbul, Turkey, May 1999),149 -157.

LEVI, A., AND CAGLAYAN, M. U. 1999b. Verification of Classical Certificates via Nested Certificates and Nested Certificate Paths. In *Proceedings of ICCCN99 – Eight International Conference on Computer Communications and Networks* (Boston, MA, Oct. 1999), 242 - 247.

LEVI, A., AND CAGLAYAN, M. U. 1999c. Analytical Performance Evaluation of Nested Certificates. *Performance Evaluation 36-37*, 213 - 232.

MASTERCARD INC. 1997. *SET Secure Electronic Transaction Specification Book 1: Business Description*.

MENEZES, A. 1993. *Elliptic Curve Public Key Cryptosystems*, Kluwer Academic Publishers.

MICALI, S. 1996. *Efficient Certificate Revocation*, MIT Laboratory for Computer Science, Technical Memo 542b.

MYERS, M. 1998. Revocation: Options and Challenges. In *Proceedings of Financial Cryptography 98* (Anguilla, BWI, Feb. 1998), LNCS 1465, Springer-Verlag, 165-171.

MYERS, M., ANKNEY, R., MALPANI, A., GALPERIN, S., AND ADAMS, C. 1999. *X.509 Internet Public Key Infrastructure On-line Certificate Status Protocol – OCSP*, RFC 2560.

NAOR, M., AND NISSIM, K. 2000. Certificate Revocation and Certificate Update. *IEEE Journal on Selected Areas in Communications 18*, 4 (Apr.), 561 - 570.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST). 1994. *Digital Signature Standard (DSS)*, Federal Information Processing Standard (FIPS) PUB 186, U.S. Department of Commerce, Washington, D.C.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST). 1995. *Secure Hash Standard (SHS)*, Federal Information Processing Standard (FIPS) PUB 180-1, U.S. Department of Commerce, Washington, D.C.

RAMSDELL, B. 1999. *S/MIME Version 3 Certificate Handling*, RFC 2632.

RIVEST, R. 1992. *The MD5 Message Digest Algorithm*, RFC 1321.

RIVEST, R. 1998. Can We Eliminate Certificate Revocation Lists?. In *Proceedings of Financial Cryptography 98* (Anguilla, BWI, Feb. 1998), LNCS 1465, Springer-Verlag, 178-183.

RIVEST, R., AND LAMPSON, B. 1996. *SDSI – A Simple Distributed Security Infrastructure*, http://theory.lcs.mit.edu/~cis/sdsi.html.

RIVEST, R., SHAMIR, A., AND ADLEMAN, L. 1978. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM 21*, 2 (Feb.), 120 - 126.

STALLINGS, W. 2003. *Cryptography and Network Security Principles and Practice, third edition, Chapter 15*. Prentice-Hall.

UNITED STATES POSTAL SERVICE. 1998. *Performance Criteria for Information-based Indicia and Security Architecture for IBI Postage Metering Systems*, http://www.usps.gov/ibip/documents/specs/pc0819.pdf.

WAP FORUM. 2001a. *Wireless Application Protocol Architecture Specification*, WAP-210-WAPArch-20010712, latest version is at http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html.

WAP FORUM. 2001b. *Wireless Transport Layer Security Specification*, WAP-261-WTLS-20010406-a, latest version is at http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html.

WAP FORUM. 2001c. *Wireless Application Protocol Public Key Infrastructure Definition*, WAP-217-WPKI-20010424-a, latest version is at http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html.

Received; revised; accepted.